
This is the **published version** of the bachelor thesis:

Alejo García, Miquel; Martí Godia, Enric, dir. "Re : Write Dungeon, un joc Rogue-Like amb generació procedural-aleatòria. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/248515>

under the terms of the  license

Re:Write Dungeon, un joc Rogue-Like amb generació procedural-aleatòria

Miquel Alejo García

Resum— Aquest article explica el disseny, implementació i test d'un prototipus de videojoc 2D pixel art amb vista zenital del gènere Rogue-Like fet amb Unity. Aquest joc, al igual que el seu gènere, es caracteritza principalment per la generació procedural-aleatòria dels elements. Els nivells es generaran de forma aleatòria en cada partida fent ús de sales prefabricades. El tipus d'enemics que confrontarà el jugador vindran decidits per un sistema que aprofiti el triangle de debilitats també conegut com la regla del **Rock-Paper-Scissors**. El joc presenta 3 accions que funcionen amb el sistema del pedra-paper-tisores. Si el jugador abusa d'alguna d'aquestes accions el joc generarà enemics resistents a aquesta mecànica segons el triangle de debilitats i d'aquesta manera afegirà balanceig al joc.

Paraules clau— Unity, Rogue-Like, generació procedural-aleatòria, Run, Boss, Gameplay, Frame

Abstract— This article explains the design, implementation and test of a 2D pixel art Rogue-Like video game prototype with a zenithal view made with Unity. This game, like its genre, will be characterized primarily by procedural-random generation elements. The levels will be generated randomly each game using prefabricated rooms. The type of enemies the player will face will be decided by a system that takes advantage of the triangle of weaknesses also known as the **Rock-Paper-Scissors** rule. The game features 3 actions that work with this system, if the player abuses any of these actions the game will generate enemies resistant to this mechanic according to the triangle of weaknesses also adding balance to the game.

Keywords— Unity, procedural-random generation, Run, Boss, Gameplay, Frame

1 INTRODUCCIÓ

En el món dels videojocs sempre s'ha treballat amb la idea d'un joc que es dissenya de forma que el jugador progressa fins a arribar al final, treballant així amb punts de guardat per a evitar tornar a l'inici en cas de morir.

El gènere **Rogue-Like** [1] juga amb una mecànica totalment diferent i trenca aquesta idea base. En aquest tipus de jocs, el jugador ha de començar des del punt inicial cada cop que mor, sense cap punt de guardat pel mig. A més, cada nova partida o 'Run' l'escenari canvia totalment. Això es així perquè els jocs Rogue-Like es caracteritzen per treballar amb la generació procedural i

aleatòria, provocant així que cada intent sigui diferent a l'anterior. Novament l'objectiu aquí també és finalitzar el joc, però aquest gènere ho proposa de manera diferent. En aquests jocs es requereix habilitat per part del jugador, però a més, juga amb la idea de que cal tenir sort per a que es generin elements favorables durant aquesta nova partida. La durada real d'una partida en un Rogue-Like no sol arribar a una hora però, el fet d'afegir-li aquesta aleatorietat i la necessitat d'habilitat per a arribar al final sense morir, provoca que el joc pugui arribar a donar més de 200h d'entreteniment.

Per a mi, els jocs Rogue-Like tenen un concepte molt divertit, que és jugar amb la aleatorietat, el que ofereix un joc diferent en cada partida. Això m'ha proporcionat moltes hores d'entreteniment, el que m'ha portat a interessar-me en aprendre a dissenyar-los i investigar com funcionen els seus algorismes d'aleatorietat. En el món dels Rogue-Like es poden trobar un munt d'idees diferents i encara que el nucli busca sempre ser el mateix, hi ha companyies

• E-mail de contacte: miquel.alejo@gmail.com

• Menció realitzada: Computació

• Treball tutoritzat per: Enric Martí Godia (Ciències de la computació)

• Curs 2020/21

que saben donar-li un toc especial al seu joc. Per a algú interessat en la programació i els videojocs aquest gènere és una bona aproximació per a introduir-se. Els Rogue-Like no són tan exigents com pot ser un del tipus AAA[2] i encara així aprendre a fer-los suposa un gran repte. Un dels seus avantatges és que no fa necessari tenir un equip de 80 o 90 persones treballant per poder realitzar-ho, el que fa que avui dia la majoria de companyies independents solen començar amb un joc d'aquest tipus. És per això que considero aquest projecte una oportunitat per aprendre el funcionament dels algorismes així com introduir-me en el món dels videojocs i més concretament en el dels motors de jocs com és en aquest cas Unity.

1.1 Estat de l'art

Les meves referències del gènere i que m'han inspirat per a dissenyar el meu joc han sigut els jocs Rogue-Like **Enter the gungeon**, i **The binding of Isaac: Rebirth**, essent aquest últim el que més ha influït. També he de comptar amb la referència de **Super Smash Bros** per al sistema de balanceig.



Fig. 1: The Binding of Isaac: Rebirth gameplay image

The binding of Isaac: Rebirth [3] (figura 1) Es un joc Rogue-Like 2D amb vista zenital desenvolupat per Edmun McMillen i publicat el 4 de novembre de 2014. Aquest joc va ser un detonant en el món dels Rogue-Like i un punt de referència per a molts altres entregues que el van succeir. El joc, encara que simple d'entendre, compta amb una gran quantitat de contingut i una gran varietat d'elements que fan que cada partida jugada sigui completament diferent. Ofereix també diferents finals i el propi joc exigeix que es superi diverses vegades per tal de desbloquejar contingut. Cada nivell es genera de manera procedural incrementant la dificultat a cada nivell. Primer es decideix quin tipus de mapa serà, el seu tamany i si té alguna maledicció (la qual provocarà un efecte negatiu com no poder veure el minimapa o no poder veure quin objecte s'agafa). Després d'això es generen totes les sales complint els criteris previs.

Enter the gungeon [4] (figura 2) És un Rogue-Like 2D també en vista zenital desenvolupat per l'equip **Dodge Roll** i publicat el 5 d'abril del 2016. En aquest cas parlem d'un joc publicat 2 anys després de la sortida d'Isaac, on una de les mecàniques que més podem apreciar és el **BulletHell** [5] la qual tracta principalment una quantitat impressionant de projectils en pantalla que el jugador ha de



Fig. 2: Enter the Gungeon gameplay image

ser capaç d'esquivar amb molta habilitat. Si be Isaac també compta en certa manera amb aquesta mecànica, no es va fer remarcable fins anys després amb la sortida de les seves expansions i és per això que no es menciona en l'apartat anterior. Enter the Gungeon presenta un objectiu similar al de Isaac: avançar pels nivells superant als enemics que trobis. No obstant en aquest cas les mecàniques que ofereix són més avançades i donen pas a un gameplay totalment diferent a l'anterior joc. Una gran diferencia és el combat, on en aquest cas es dispara utilitzant el ratolí per apuntar i no està limitat a les 4 direccions base com és el cas d'Isaac. Una part interessant i que s'ha utilitzat com a referència per al projecte, és el fet de que les sales no estan unides per salts de pantalla sinó que estan connectades per passadissos oferint així continuïtat en el moviment. A més la càmera segueix al jugador en comptes de estar fixa a cada sala.



Fig. 3: Super Smash Bros ultimate cover image

Super Smash Bros [6] (figura 3) Concretament parlaré de la seva última edició "Super Smash Bros ultimate", desenvolupada per Masahiro Sakurai i publicat el 7 de desembre de 2018. Parlem d'un joc completament de lluita que engloba com a personatges els més icònics de les franquícies de **Nintendo**, on la idea principal és lluitar amb un sistema de combat típic d'aquest tipus de jocs en diferents modalitats que ofereix el joc. La part interessant i que el fa per a mi una referència és com funciona el sistema de pedra-paper-tisores amb la IA del joc. Encara que compta amb una gran varietat d'habilitats, es poden denotar tres accions principals: atacar, agafar a l'adversari i bloquejar. Aquestes tres accions compleixen el triangle de debilitats, sent l'atac superior a agafar, bloquejar més que atacar i agafar superior a bloquejar. La gràcia d'aquest sistema, és que a mesura que es lluita contra la IA del joc,

aquesta aprèn i té en compte els moviments que utilitzes. Si el jugador abusa massa del fet de només atacar, la IA ho aprendrà i començarà a bloquejar els atacs que el jugador faci, funcionant per a les altres dues accions també. Aquesta idea provoca que el jugador es vegi obligat a aprendre a lluitar i tenir un combat equilibrat, decidint què és millor fer en cada moment.

Avui dia els Rogue-Like estan molt present en el món dels videojocs, i si bé són lluny de ser jocs AAA, són un dels gèneres més populars gràcies al repte que suposa per als usuaris a el fet d'oferir una experiència de joc totalment diferent a la habitual. Dissenyar un bon Rogue-Like no suposa un gran cost de recursos, siguin econòmics o de software. És per això que aquest tipus de jocs permeten a companyies petites, o que acaben de començar, desenvolupar un primer títol que ben fet pot donar una bona empenta en aquest món. Aquest tipus de jocs demostren que no fa falta tenir l'etiqueta de AAA per a poder ser un gran joc i deixar marca.

2 OBJECTIUS DEL PROJECTE

L'objectiu principal del projecte és dissenyar i desenvolupar un prototipus de videojoc 2D estil pixel art amb vista zenital del gènere Rogue-Like amb les tasques i objectius següents:

- Dissenyar i implementar un algorisme capaç de fer generacions de sales aleatòries però amb sentit i continuïtat. Això vol dir que el joc ha de ser capaç de generar un nivell amb sales que es connectin correctament entre elles i generar sempre una sala final on ubicar el **Boss** per tal de poder acabar el nivell. El disseny de cada nivell haurà de ser sempre aleatori.
- Dissenyar i implementar un sistema de balanceig que actuï en funció de les accions del jugador. El sistema es dissenyarà seguint el triangle de debilitats conegut també com les regles del pedra-paper-tisores (figura 4).
- Crear un sistema de combat i moviment funcional i jugable que no generi conflicte amb la resta d'elements del joc.
- Dissenyar 3 tipus d'enemics amb característiques i mecàniques pròpies i diferenciables pel jugador.
- Aprendre l'ús del motor Unity així com les seves interaccions amb el llenguatge de programació C#
- Aconseguir una versió jugable sense errors que tingui un principi i un final de nivell.

Les eines software utilitzades per a l'elaboració del projecte són:

- **Motor Unity:** Serà l'entorn on s'ha desenvolupat el gameplay del joc.
- **Visual Studio amb configuració de llenguatge C#:** Tota la programació necessària pel joc s'ha treballat en llenguatge C#, el més eficient per treballar amb Unity.

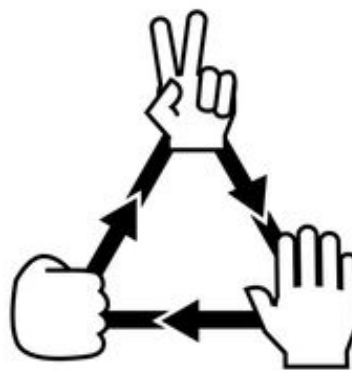


Fig. 4: Rock Paper Scissor System

- **Aseprite:** Entorn de dibuix pixel art utilitzat per al disseny de tot l'art del joc. Aquest programa s'especialitza en el disseny de pixel Art així com animacions per aquest.
- **Trello:** Eina utilitzada per a la gestió i seguiment de tasques a realitzar durant tot el projecte.

2.1 Metodologia

Per a dur a terme aquest projecte s'ha implementat una versió reduïda i simplificada del que seria la metodologia **Scrum**. La decisió ha sigut Scrum per la seva eficiència en el treball iteratiu. He fet ús de l'eina **Trello** (figura 5) que per a gestionar un projecte essent una persona resulta molt còmode.

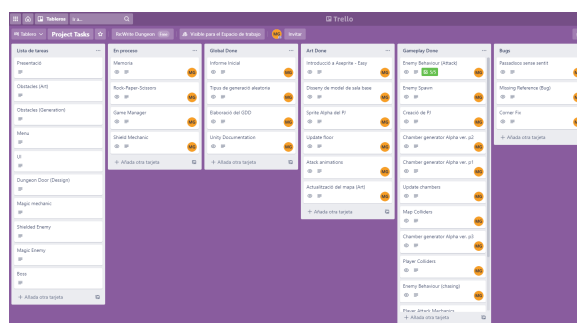


Fig. 5: Taula Trello on es mostren les tasques del projecte

La idea era treballar setmanalment en unes tasques concretes, que decidia i revisava a l'inici de cada setmana. Al començar el projecte es van crear a Trello les tasques principals. Durant el projecte i a mesura que es trobaven nous reptes les tasques s'anaven tornant més complexes. Al no haver treballat mai en el disseny real de un videojoc era difícil entendre des d'un principi quines eren totes les tasques necessàries. Com es va preveure, a mesura que el projecte avançava es van anar descobrint altres tasques importants que, en un principi, s'havien passat per alt, a més de tasques que es podien desglossar en altres per a una fàcil estructuració. Cada dilluns revisava quines tasques podia donar per acabades de forma definitiva, quines noves

podia afegir segons el progrés del projecte i si feia falta crear alguna checklist d'una tasca que creia més gran del que en un principi havia estimat.

Durant la elaboració del projecte durant la setmana, sigués treballant o en hores lliures, era habitual pensar alguna idea per al joc o la implementació de segons quin algorisme. En aquest aspecte Trello ha resultat molt útil donat la fàcil accessibilitat i ràpid maneig que, amb l'afegit de la seva aplicació mòbil, em permetia afegir tasques noves en qualsevol moment. Arribat el dilluns podia revisar si aquestes idees eren factibles de treballar. Amb l'ús d'aquesta eina, he pogut treballar de manera estructurada sense desbordar-me de treball en cap moment.

Els continguts d'aquest article es distribueixen en els següents apartats:

- **Secció 3 Game Design Document:** Informació sobre el videojoc realitzat.
- **Secció 4 Art:** L'art que s'ha dissenyat per al videojoc.
- **Secció 5 Desenvolupament:** Explicació i implementació dels mòduls del projecte.
- **Secció 6 Resultats:** Visió final sobre el videojoc resultant.
- **Secció 7 Conclusions i Millores:** Conclusions sobre el videojoc final i les seves possibles millores.

3 GAME DESIGN DOCUMENT

Encara que per aquest treball es presenta com un prototipus, com a joc Re: Write Dungeon busca ser un joc de gènere Rogue-Like 2D amb ambientació Dark Fantasy totalment dissenyat en pixel art. L'objectiu del joc serà avançar a través dels diferents nivells generats aleatòriament mentre el jugador combina les diferents accions de combat que el joc ofereix per tal de mantenir un balanceig d'enemics en cada sala. Cada sala registrarà el nombre de vegades que el jugador utilitza una acció. Si abusa massa d'alguna d'elles, els enemics de la següent sala seran majoritàriament resistents a aquesta acció. Com ampliacions futures de l'estat actual, el joc disposarà de diferents nivells amb diferents estils. Cada nivell augmentarà la dificultat. A cada nivell es podran anar trobant objectes que modificaran l'estat actual del jugador, donant-li més atac, més velocitat, donar-li més rang d'atac, etc. La durada de cada nivell és de entre 3-6 minuts. La durada sempre depèn de saber lluitar contra els enemics i tenir la sort de escollir el camí correcte.

3.1 Estil de l'art

L'estil serà pixel art amb una intenció **Dark Fantasy**. Aquest estil es caracteritza per una ambientació medieval fantàstica on tot l'entorn i enemics solen ser de caràcter obscur i tèntric. S'ha pres la decisió de pixel art doncs és una de les més senzilles d'aprendre i que pot arribar a donar grans resultats en mans de professionals.

3.2 Com es juga?

El jugador ha avançar per les sales derrotant enemics fins a trobar la sala del **Boss**, lluitar amb ell i donar per finalitzat el nivell. El jugador es pot moure amb les tecles **W**, **A**, **S**, **D** i atacar amb la barra espaiadora. Per protegir-se cal mantenir la tecla **Q** i per llençar màgia s'ha de prémer la tecla **E**. Cal remarcar que tant la màgia com l'escut no es poden fer servir en moviment. Això provoca que el jugador hagi de decidir quan i com es millor utilitzar aquestes dues accions. El jugador podrà executar 3 accions a part del moviment: atacar, bloquejar o màgia. Cada acció predomina per sobre d'un altre. L'atac és més ràpid que la màgia i predomina contra els enemics que fan ús d'aquesta. D'altre banda no pot fer mal als enemics amb escut. La màgia és lenta però pot fer mal sense problemes als enemics amb escut encara que no pot fer front contra els atacs ràpids. L'escut pot bloquejar els atacs no màgics en qualsevol direcció.

A la vegada que el jugador compta amb aquestes accions, existiran 3 tipus d'enemics, cada un especialitzat en una acció. Per a fer un combat balancejat el joc tindrà en compte el número de vegades que el jugador utilitza una acció. A l'acabar una sala, amb aquests valors registrats, el joc decidirà si el jugador ha abusat massa d'alguna de les 3 accions per sobre de les altres. D'aquesta manera, si n'hi ha alguna, a la següent sala es generaran diversos enemics on la major part seran del tipus contrari, (p.e si el jugador abusa de l'atac, la següent sala contindrà majoritàriament enemics amb escut).

El jugador es mourà a través de les diferents sales fins a trobar la sala on és el **Boss**. Si es topa amb una sala sense sortida, el jugador haurà de tornar enrere i provar un altre camí. Només hi haurà enemics a cada sala nova i aquestes no es reinicien si el jugador torna a una ja vista. Per acabar el nivell, el jugador haurà d'arribar a la sala del **Boss** i superar-lo.

Vaig escollir el nom de **Re:Write Dungeon** amb la idea de enfocar el títol al nucli del joc i del projecte, la generació de nivells que sempre son diferents, el que dona la idea d'una masmorra que es re-escriu tota la estona. Aquest nom, en cas de continuar amb el disseny i implementació del joc en un futur, pot donar pas a una bona historia entorn al fet de que es re-escriu cada cop que s'entra. Com a toc de disseny vaig decidir el "Re:" que dona un estil diferent de lo habitual i que sempre m'ha semblat curiós el seu ús en títols d'animacions japoneses.

4 ART

M'he decantat per utilitzar pixel art. Com el seu nom indica, aquest estil s'enfoca en el disseny pixel a pixel per a dissenyar tot tipus d'objectes i paisatges. He escollit pixel art doncs a més d'interessar-me, m'oferia la possibilitat d'aprendre sobre l'estil i podia implementar-ho en el meu propi joc. Aquest tipus d'art pot arribar a ser molt complexe depenent del grau de detall que es busqui, però aprendre els conceptes bàsics és relativament senzill. Així doncs, escollir aquest estil em permetia fer el meu propi art. Per al disseny de pixel art s'ha utilitzat l'eina **Aseprite** [8]. La font de text del joc que actualment és només utilitzada

per als números de la barra de vida i escut és l'únic element agafat d'una llibreria gratuïta oferida pel mercat propi de Unity: **Free Pixel Font - Thaleah** [9]

4.1 Disseny del nivell

Quan parlem de nivells parlem concretament del disseny de les sales, doncs són aquestes les que composen el nivell. A continuació es mostra tot l'art referent a aquestes.

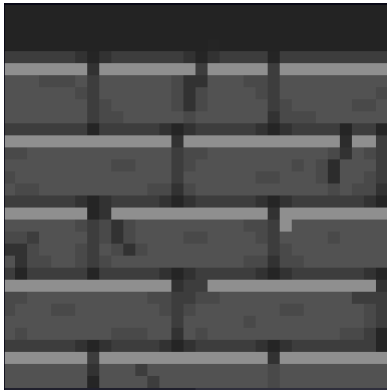


Fig. 6: Disseny de mur: 32 x 32px

El disseny de murs (figura 6) va ser el primer en realitzar-se. Es van decidir uns murs grisos amb esquerdes per donar un estil més obscur. Es va fer un disseny pensant en la repetició de la textura al llarg de la sala. D'aquesta manera els maons estan dibuixats de manera que la unió de diferents murs doni una sensació de continuïtat sense talls.

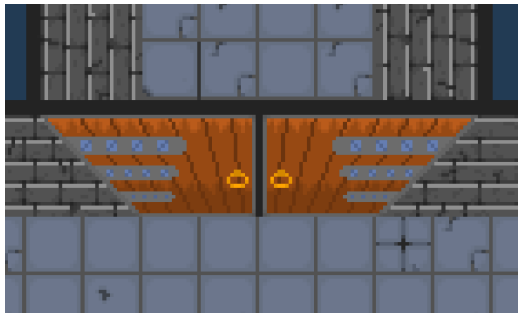


Fig. 7: Disseny de porta: 32 x 32px

Més endavant es van dissenyar les portes (figura 7) i un terra (figura 8) que acompanyés l'estètica desgastada dels murs. Es va decidir en aquest cas per una tonalitat més blava per així no donar més color als nivells a més de diferenciar clarament el terra dels murs.

Finalment es van dissenyar els passadissos, afegir les textures i afegir les portes de cada sala (figura 9). Això va provocar l'augment de tamany de 320 x 320 px al doble, 640 x 640 px. Cada sortida compta amb un passadís de 160px que s'unirà amb el passadís de les sales contigües.

4.2 Personatges

Com a personatges vaig decantar-me per un disseny senzill que no donés massa complicació doncs l'art no és el tema principal del projecte.

Per al disseny de Alpha, el personatge que el jugador controlarà (figura 10), no vaig prendre cap referència més

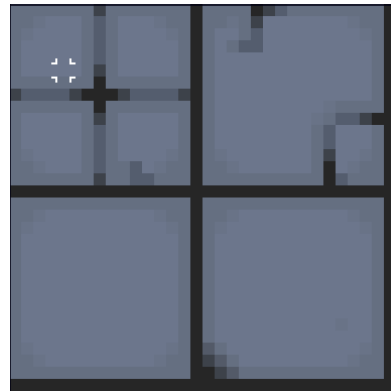


Fig. 8: Disseny de terra: 120 x 32 px

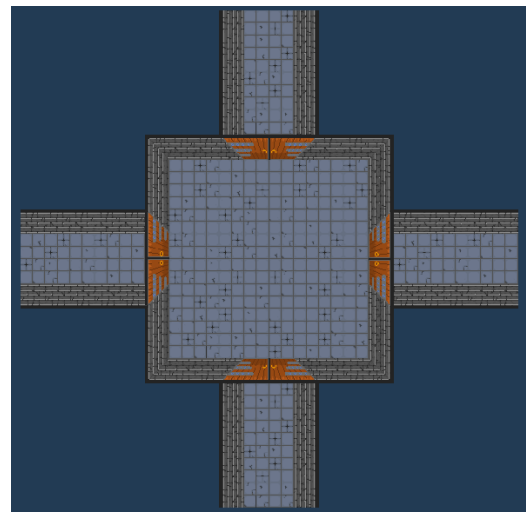


Fig. 9: Sala central: versió final amb passadissos 640 x 640px



Fig. 10: Alpha, personatge principal del joc

que anar fent proves a l'editor de Aseprite fins que va sortir un disseny que em convencia. L'aparença és un símil a un **Slime** un monstre de baba molt típic dels jocs de fantasia. Vaig decidir aquest disseny per a evitar complicar-me amb les animacions de moviment doncs un enemic amb cames i sense animació es veu massa forçat a l'escenari.

Seguidament, per a fer els enemics, buscava un disseny que fos fàcil de dur a terme i que identifiqués clarament que era un enemic. Per aquest motiu, vaig decidir utilitzar el model del personatge principal i modificar els colors i tonalitats d'ombres per a crear el primer enemic: **Beta** (figura 11).

Aquest enemic, al igual que Alpha, té les proporcions de 32 x 32 píxels ocupant la totalitat d'aquest a cada banda. Fent ús d'aquest model es van dissenyar els dos enemics restants. El **Shielder** (figura 12) agafa la idea del disseny de beta afegint un escut davant seu. En els dissenys laterals, l'escut està dissenyat de manera que es noti que existeix encara que estigui de perfil.



Fig. 11: Beta, enemic tipus atac



Fig. 12: Shielder, enemic tipus defensiu



Fig. 13: Wizard, enemic tipus magic



Fig. 14: Boss del nivell

El disseny del **Wizard** (figura 13) és possiblement el més diferenciat de la resta d'enemics. En aquest cas em vaig decantar per un disseny més creatiu treballant en el seu barret.

Es va decidir que el disseny del **Boss** (figura 14) seria totalment diferent per a provar noves idees. Aquest és l'únic enemic amb un tamany superior, 64 x 64 px.

5 DESENVOLUPAMENT

En aquest apartat es mostra com s'ha estructurat el gameplay. En la figura 15 es mostra un diagrama de mòduls dels elements més importants del joc i com estan connectats entre ells.

La mecànica general és la següent: El joc connecta amb l'usuari mitjançant una GUI (*Graphic User Interface*). El jugador rep informació i interactua amb el joc directament per aquest mòdul. El joc es pot diferenciar en dos apartats, l'entrada d'informació i la visualització. El primer s'encarrega de rebre les accions del jugador en forma de dades. En aquest apartat els mòduls involucrats són tots aquells que funcionen en base a les interaccions del jugador. D'altra banda la part de visualització engloba tots els mòduls que no depenen en cap moment de les accions de l'usuari, són mòduls que modifiquen el joc per si mateixos i mostren aquests canvis via la GUI. Seguidament s'explicaran les principals funcionalitats de cada mòdul.

5.1 GUI (Graphic User Interface)

Interfície gràfica que l'usuari visualitza a través de la seva pantalla. Es mostra tota la informació i permet a l'usuari interactuar amb el joc. A més del joc es mostren constantment la barra de vida i la barra d'escut.

5.2 Entry

Engloba tot l'apartat interactiu que té el joc. Aquest mòdul rep les instruccions de l'usuari que són enviades al mòdul de personatge, el qual actuarà en funció de les dades rebudes.

5.3 Visualization

Torna en format visual totes les dades que el joc ha hagut de processar en cada frame. En aquest grup es situen principalment tots els elements que funcionen de forma independent a les accions del jugador, encara que això no vol dir que no interactuin entre ells.

5.4 Player Behaviour

El jugador interactua amb el joc mitjançant el personatge. És per aquest motiu que el mòdul d'entrada infereix directament en aquest. El personatge té dos comportaments/interaccions: el moviment i el combat. El jugador té moviment lliure en les quatre direccions: amunt, abaix, esquerra i dreta, a més de les respectives diagonals. D'altra banda dins el combat el jugador pot: atacar, bloquejar i llençar màgia.

- **Atac:** EL jugador pot fer un cop senzill en una de les 4 direccions. És més ràpid que l'ús de màgia. Eficaç

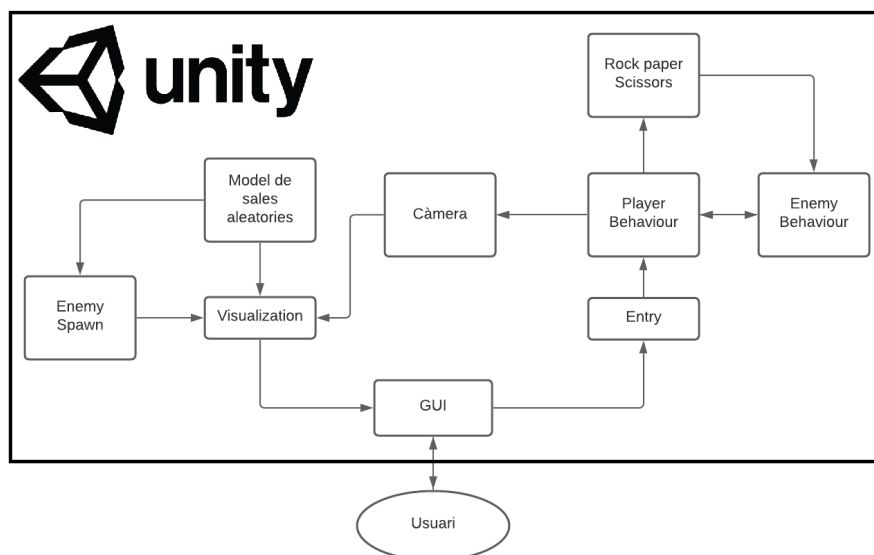


Fig. 15: Diagrama de mòduls del joc Re:write Dungeon

contra enemics màgics, inútil contra enemics amb escut. Atacar a un usuari de màgia permetrà atacar i moure abans de que el llançament de la màgia acabi.

- **Escut:** EL jugador pot bloquejar els atacs no màgics entrants mantenint polsant un botó. L'escut té un màxim de punts de resistència, cada cop que rep un cop es perden punts, es regeneren amb el temps mentre el jugador no l'estigui utilitzant. Si l'escut perd tota la resistència es trencarà i trigarà un temps més extens en recuperar-se. La màgia pot travessar sense problemes aquest escut.
- **Màgia:** La màgia compta amb un rang d'atac una mica més llarg però triga més en activar-se el que provoca que sigui més fàcil rebre atacs ràpids de l'enemic. D'altra banda la màgia pot travessar perfectament l'escut enemic independentment d'on es trobi.

El codi encarregat de fer els atacs necessita sempre una direcció per a poder funcionar. Això es així per poder identificar sempre en quina direcció ataca el jugador sense atacar les quatre a l'hora.

5.5 Enemy Behaviour

Tots els enemics tenen un comportament agressiu contra l'usuari. Això ve determinat per un algorisme de chasing que segueix en tot moment al jugador. Addicionalment, s'utilitza l'eix de coordenades Z per a provocar una reducció de velocitat de moviment a mesura que els enemics s'apropen al jugador. El combat funciona de manera independent al moviment. Cada enemic té un detector d'objectes a cada costat. Si el jugador entra en un dels 4 detectors, s'activarà l'algorisme d'atac (diferent depenent del tipus d'enemic).

5.6 Rock paper Scissors

Aquest mòdul és un dels dos més importants del projecte. Com a toc addicional per al joc, es va decidir elaborar un sistema, a més de la generació de sales, que afectés al transcurs de la partida. Es va pensar, doncs, en el triangle de

debilitats conegut també com **Rock-Paper-Scissors rule**. Aquest sistema és utilitzat en molts jocs de diverses maneres però sempre amb la mateixa idea: establir diferents rols que estableixin fortaleces i debilitats per a fer pensar al jugador. La idea va sorgir pensant en el sistema de combat de la IA del joc **Super Smash Bros ultimate**, ja explicada anteriorment.

L'algorisme per a implementar aquesta mecànica funciona sempre depenent del jugador. La sala inicial sempre contindrà enemics aleatoris. No obstant des d'aquí, es comencen a comptar numèricament totes les accions del jugador sumant un a un cada cop que les reutilitza. Una vegada finalitzada la sala el jugador avança a la següent. Quan entra a la nova sala, un **trigger** s'activa just a l'entrada, el codi inicia i rep tota la informació acumulada sobre les accions que el jugador porta a sobre. El algorisme comprova que d'entre les 3 accions, la més utilitzada no supera en un valor fixat a la segona més utilitzada. Si això succeeix el algorisme indica al joc quin tipus de plantilles (figura 16) d'enemics ha de fer aparèixer.

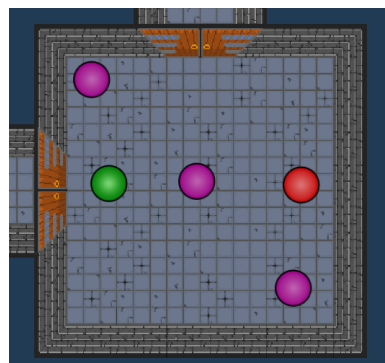


Fig. 16: Sala amb plantilla enfocada a generar mags (Punts morats).

Aquestes plantilles estan distribuïdes segons si son plantilles d'enemics variats o plantilles que prioritzen un enemic en concret. S'utilitza aquest sistema per afegir aleatorietat inclòs en situacions de balanceig perquè d'aquesta mane-

ra el jugador no pugui preveure quins enemics exactament apareixeran sabent que ha abusat d'una de les tres accions disponibles. Amb aquestes mecàniques es busca que el jugador sàpiga fer un ús equilibrat de les seves accions.

5.7 Càmera

La càmera és el mòdul pont entre els dos apartats que engloben la GUI. La càmera segueix en tot moment la posició del jugador, fent que aquest sigui sempre el centre de la pantalla. La càmera funciona com un objecte independent amb un codi que busca en tot moment la posició del jugador. La pròpia càmera és qui acaba mostrant tot el contingut visual del joc delimitant què es veu i què queda fora de pantalla. La pròpia càmera portarà sempre de forma estàtica el HUD[7] (*Heads-up display*) on per aquesta versió es mostra la barra de vida i l'escut.

5.8 Model de sales aleatòries

L'altre nucli del joc i amb més pes en el projecte ha sigut dissenyar un algorisme capaç de crear nivells de masmorra amb combinacions aleatòries de sales prefabricades. Aquest algorisme és capaç de crear nivells que tinguin sentit, utilitzar les sales adequades per a cada posició, crear camins que no es trenquin, establir un punt final de nivell donant la possibilitat de finalitzar el nivell i tot això de forma autònoma sense la necessitat d'interacció del usuari.

Per al joc, s'ha decidit l'opció de crear un nivell a partir de sales prefabricades que jo mateix he dissenyat. Si bé un primer pensament pot ser que la prefabricació és fer trampes, s'ha de entendre que no existeix pràcticament cap *Rogue-Like* que no porti aquesta prefabricació. Tots aquests jocs necessiten tenir un mínim patró per poder decidir un camí fins al final del nivell. Establir sales prefabricades, permet dissenyar un algorisme que tingui en compte què cal utilitzar segons la situació en la que es trobi.

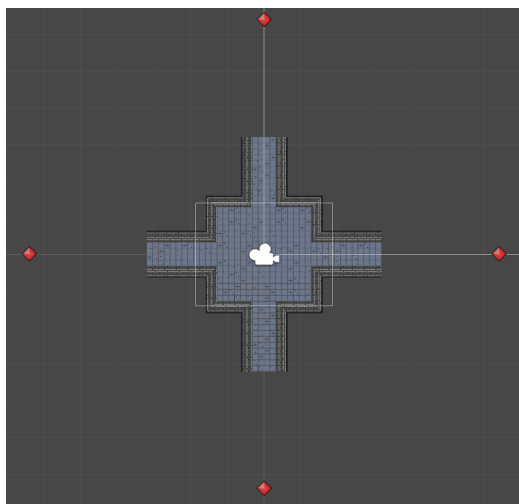


Fig. 17: Sala principal amb passadissos i punts de generació de sales

A cada sortida a certa distància de la sala actual s'estableix un punt d'aparició que conté el codi per a generar sales (figura 17). Aquests punts es situen a una distància exacte, donat que representen el centre d'on apareixerà la nova sala i per tant s'ha de evitar que es solapin o que deixin espais

buits. L'algorisme anirà incorporant sales fins que la sala generada no tingui sortida el que provocarà un final de camí.

per evitar complicacions d'execució s'ha afegit un petit retràs en segons a la creació de cada sala per a que no es doni el cas de execucions de codi al mateix exacte instant de temps. S'ha implementat una llista que registra totes les sales que es van generant. Aquesta llista permet a l'algorisme identificar quina és l'última sala generada, així es pot marcar aquesta com la sala del Boss, solucionant el problema d'un nivell sense final. S'ha afegit a més un limitador de sales que provoca que a partir de 10 sales l'algorisme comença a posar a les sortides restants sales sense sortida. s'ha dissenyat així doncs com la generació de sales és simultània en diferents direccions és bastant difícil establir un número de sales exacte. Per que s'entengui, si tenim dos sortides i el nombre de sales actual és 9, una sortida generarà la 10 i ho comunicarà a una variable global però pot ser que la sortida restant comenci a executar-se abans que la primera acabi i per tant no detecti que hi ha 10 sales el que provoca que la següent sala seguirà els patrons habituals.

Per a solucionar un problema de l'algorisme que generava sortides que apuntaven a parets d'altres sales deixant camins oberts es va decidir implementar un nou punt de detecció. Aquests punts, identificats a la figura 18 com a punts

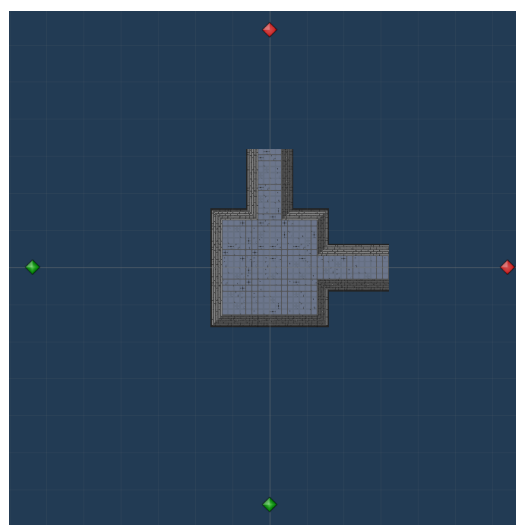


Fig. 18: Sala cantonada amb detectors de murs i punts de sortida

verds, indiquen quin mur estan tocant, dividint-se així en punts de paret esquerra, dreta, amunt i baix. Llavors quan un punt de generació (punt vermell) detecta a la seva posició un punt de detecció, rep la informació sobre quins murs té a prop i en quina direcció estan. Amb aquesta nova informació, l'algorisme es modifica per a seleccionar només un tipus de sales. Per a ser més clar, dins el codi s'eliminen totes les sales amb sortides que apuntin a una paret. Per al problema de les cantonades que es destrüïen s'ha afegit un detector que instancia un obstacle en ambdós sortides per que el jugador identifiqui que són camins sense sortida. Per a afegir més aleatorietat respecte al joc, s'ha decidit implementar en l'entrada de cada sala, un sistema que esculli una plantilla de punts d'aparició d'enemics aleatòria que s'activa cada cop que el jugador entra per primer cop a una sala.

5.9 Enemy spawn

L'aparició dels enemics ve definida per un set de plantilles que s'ubiquen en diferents parts de la sala un punt d'aparició. Aquesta plantilla s'encarrega de col·locar aquests punts d'aparició, els quals seguidament activen el codi per a fer aparèixer l'enemic decidit. Aquestes plantilles es seleccionen de manera aleatòria pel mòdul de sales aleatòries. Aquest sistema permet al jugador no només jugar nivells diferents sinó la possibilitat de afrontar un número d'enemics diferents sempre. D'aquesta manera el jugador no pot predir amb certesa la situació de la següent sala, doncs encara que pugui suposar quin tipus d'enemic predominarà, no podrà saber el nombre d'enemics que apareixeran.

6 RESULTATS

Una vegada finalitzats tots els objectius que es volien implementar en el joc, s'ha construït una versió executable del joc per a poder ser testejat. Després de diferents proves i de rebre l'opinió de usuaris externs els resultats han sigut:

Es té un algorisme de generació procedural-aleatòria que genera nivells finits, capaç d'evitar trencaments del nivell, omplir espais buits, definir un punt final amb un Boss i que permet el funcionament de cada sala generada de forma independent, ja que cada una d'aquestes té els seus propis elements. Tot això de forma autònoma executant-se al iniciar cada nivell i independentment de lo que el jugador faci. Tots els nivells jugats als tests han sigut diferents. Cap de les partides jugades pels testers ha donat error d'execució i tampoc s'ha trobat cap bug.

El sistema de combat és actualment rígid i falta ser més fluït. No obstant respon correctament a les tecles i s'executen totes les accions tal i com han sigut programades. El combat funciona tal com s'esperava per a un prototip.

El sistema de balanceig funciona complint sempre amb les condicions proposades. Fa falta ajustar el nombre de vegades que una mecànica s'utilitza per a que compti com abús. Tots els usuaris que han jugat han provocat satisfactòriament el tipus d'enemic corresponent a l'abús d'una mecànica.

En l'apèndix secció A.1 es mostren imatges sobre diferents generacions de nivells.

7 CONCLUSIONS I MILLORES

Durant la elaboració d'aquest projecte he après bastant de diferents camps i com a programador he après moltes coses noves en l'àmbit dels videojocs.

- s'ha dissenyat i implementat un prototipus de joc del gènere Rogue-Like funcional.
- S'ha dissenyat i implementat satisfactòriament un algorisme de generació procedural-aleatòria de nivells
- S'ha dissenyat i implementat satisfactòriament un sistema de balanceig de joc totalment propi fent ús dels coneixements adquirits.
- S'ha dissenyat i implementat un prototipus capaç de rebre ampliacions per a ser un joc complet.

- He sigut capaç de dissenyar un art des de zero i completament propi el qual s'ha aplicat al joc satisfactòriament.
- S'han utilitzat els coneixements apresos sobre programació per poder dissenyar un codi ampliable que pot ser reutilitzat per a projectes futurs.
- S'ha aconseguit el repte de fer un sistema de col·lisions funcional. Un dels aspectes més importants de desenvolupar un videojoc.
- He après a treballar amb el moviment d'objectes que no són controlats per l'usuari.
- La carrega de treball ha sigut ben distribuïda i ben organitzada durant l'elaboració del treball.

En l'estat actual del joc i amb les opinions rebudes dels diferents testers, s'ha conclòs que les possibles millores més destacables a continuar des del punt on es troba actualment el joc són:

- Afegir animacions més llargues i fluides que donaran més vida al joc.
- S'han d'ajustar les col·lisions dels enemics i el jugador per a que el combat sigui més fluït i divertit.
- S'ha d'ajustar el rang dels atacs tant dels enemics com del jugador. A més s'ha de fer que concordi correctament amb les animacions futures. això farà que sigui més satisfactori lluitar.
- El rang d'atac del jugador es pot ampliar donant pas a millors animacions i més marge d'atac.
- Implementar obstacles a les sales fent ús del sistema de plantilles actuals. D'aquesta manera les sales ja no es veuran buides.
- Afegir fases al combat del boss. Es planeja fer que cada certs punts de vida, es divideixi en dos versions més petites d'ell. Això afegeix mecàniques noves que entretindran més al jugador.
- afegir barra de vida de boss i que aparegui quan s'entra a la seva sala.
- ampliar els tipus de sales que l'algorisme pot utilitzar (més grans, més llargues, sales amb forma de L)
- Afegir un menú de inici, una pantalla de Victorià i una pantalla de derrota.
- afegir un mètode per poder pausar el joc i reanudar-lo quan faci falta.

Aquest projecte ha sigut una experiència molt interessant per introduir-me en el camp del desenvolupament de videojocs (encara que a petita escala), el camp de la generació procedural i en el món del pixel art.

AGRAÏMENTS

Vull agrair als meus pares per tot el suport que m'han donat durant aquest treball. Als meus amics en especial la meua amiga, qui ha més d'oferir-me suport, ha fet possible el disseny del pixel art deixant-me un dispositiu de disseny i explicant-me com utilitzar-lo.

Per ultim agrair l'esforç del meu tutor qui ha estat constantment ajudant-me i donant consells per a millorar aquest projecte fins al final.

REFERÈNCIES

- [1] RogueLike, (Last update: juny de 2021) Wikipedia. Recuperat de:
<https://en.wikipedia.org/wiki/Roguelike>
- [2] AAA Video game industry (Last update: maig de 2021) Wikipedia. Recuperat de:
[https://en.wikipedia.org/wiki/AAA_\(video_game_industry\)](https://en.wikipedia.org/wiki/AAA_(video_game_industry))
- [3] Binding of Isaac: Rebirth Wiki (S.F.) Fandom. Recuperat de:
https://bindingofisaacrebirth.fandom.com/wiki/Binding_of_Isaac:_Rebirth_Wiki
- [4] Enter the Gungeon (S.F.) Dodge Roll. Recuperat de:
<https://dodgeroll.com/gungeon/>
- [5] Bullet Hell (S.F.) Giant Bomb. Recuperat de:
<https://www.giantbomb.com/bullet-hell/3015-321/>
- [6] Super Smash Bros ultimate web site (2018) Nintendo, Recuperat de:
https://www.smashbros.com/es_ES/
- [7] Heads-Up display, video games (Last update: maig de 2021) Wikipedia. Recuperat de:
[https://en.wikipedia.org/wiki/Heads-up_display_\(video_games\)](https://en.wikipedia.org/wiki/Heads-up_display_(video_games))
- [8] Aseprite Web site (2013) David Capello, Recuperat de: <https://www.aseprite.org/>
- [9] Free Pixel Art Font - Thalea (21 de febrer de 2019) Tiny Worlds, Recuperat de:
<https://assetstore.unity.com/packages/2d/fonts/free-pixel-font-thaleah-140059>
- [10] Unity Technologies. (s. f.). Unity User Manual. Recuperat de:
<https://docs.unity3d.com/Manual/index.html>
- [11] Blackthornprod. (13 Març de 2018). RANDOM DUNGEON GENERATOR - EASY UNITY TUTORIAL. YouTube. Recuperat de:
https://www.youtube.com/watch?v=qAf9axsyijY&ab_channel=Blackthornprod
- [12] Stack Exchange Inc. (s. f.). Stack Overflow - Where Developers Learn, Share, Build Careers. Stack Overflow. Recuperat de:
<https://stackoverflow.com/>
- [13] Continisio, C. (2 abril de 2020). Choosing the resolution of your 2D art assets. Unity Technologies Blog. Recuperat de:
<https://blogs.unity3d.com/es/2018/11/19/choosing-the-resolution-of-your-2d-art-assets/#:~:text=When%20importing%20graphics%20as%20Sprites,scaled%201%2C1%2C1.>
- [14] Brackeys. (15 desembre de 2019). MELEE COMBAT in Unity. YouTube. Recuperat de:
<https://www.youtube.com/watch?v=sPiVz1kfEs&t=728s>
- [15] Blackthornprod. (5 juliol de 2018). HOW TO MAKE 2D MELEE COMBAT - EASY UNITY TUTORIAL [Vídeo]. YouTube. Recuperat de:
https://www.youtube.com/watch?v=1QfxdUpVh5I&t=106s&ab_channel=Blackthornprod
- [16] Hundred Fires Games Youtbe channel (24 de març de 2019) Hundred Fires Games. Youtube. Recuperat de:
<https://www.youtube.com/channel/UCioSma02TDQUqeGVt1crpTw/about>
- [17] w3schools web site (s.f.) w3School. Recuperat de:
<https://www.w3schools.com/about/default.asp>

APÈNDIX

A.1 Mostres de nivells aleatoris

Les figures mostrades a continuació són nivells del prototipus final generats de forma aleatòria per l'algorisme implementat. Es pot observar com el tamany i direcció varia en cada imatge. Els punts vermells són els punts d'aparició on es creen les sales. Els punts verds són els detectors de murs explicats anteriorment. Remarcar que aquests dos tipus de punts no són visibles per al jugador durant la partida.

La figura 21 és un exemple de com s'han tractat les situacions en que dos sales apunten en la mateixa direcció. La solució ha sigut bloquejar la possibilitat d'aparició de sala i aplicar un disseny de passadís derruït que no deixa passar al jugador.

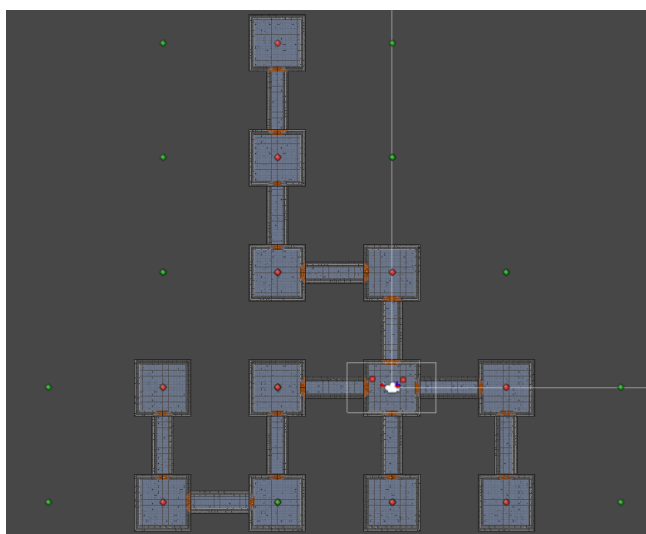


Fig. 19: Mostra 1 de nivell final generat proceduralment.

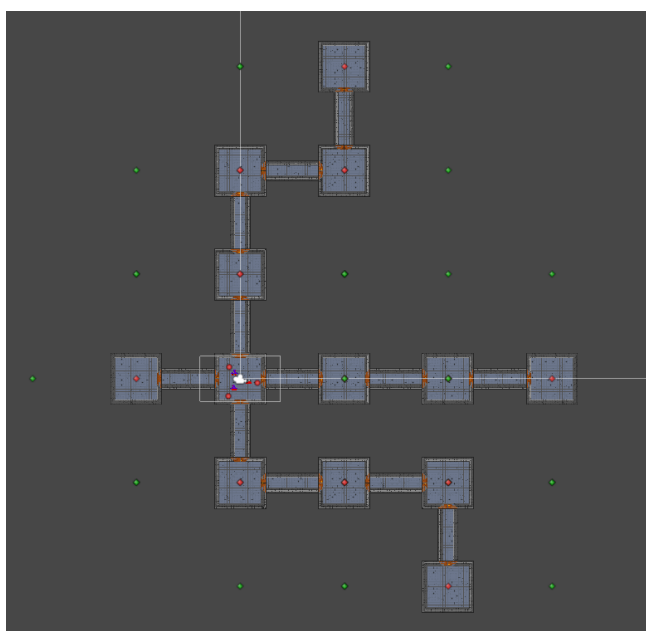


Fig. 20: Mostra 2 de nivell final generat proceduralment.

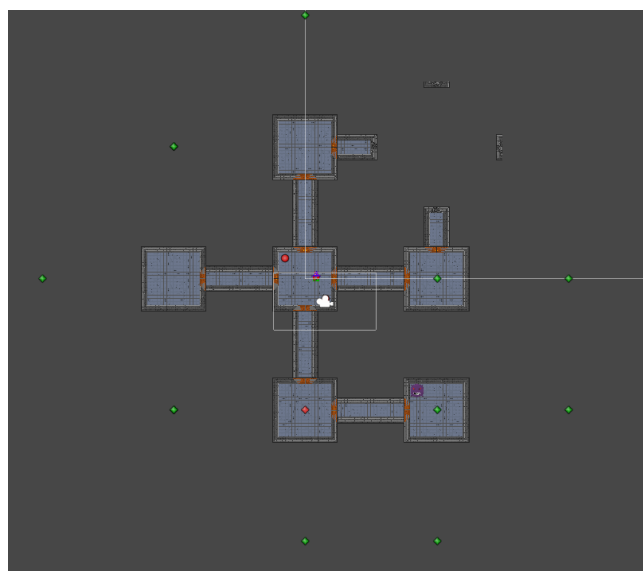


Fig. 21: Mostra 4 de nivell final amb bloqueig de camins.